

## REMARKS

Reconsideration of the objection of the drawings is requested. The suggestion that a stack has structure is subject to reconsideration. A stack is simply an area of memory. There is nothing that need be or should be shown in connection with an area of memory. It is not clear what is meant by the reference in the office action to a call stack, which does not appear to be present. There is no reason why any detail of such a stack, if there even were one, would be essential to proper understanding. Namely, a stack is simply an area of memory and, short of showing cells and structure of memory, which one skilled in the art certainly should be able to deduce on their own, there would be nothing pertinent to show.

Similarly, the objection to the specification should be reconsidered. There is nothing unclear about the language or explanation of specification. The questions posed in the office action are irrelevant. Certainly, anyone can ask infinite questions about how to do an infinite number of things, but the issue is whether any of those things are in any way relevant or necessary to understand an invention. The subject lines are apparently the language beginning at page 5, line 28. It states that after a write request is detected and if the operating system disk driver has not been loaded, as illustrated in diamond 20, then the write request is logged as indicated in block 225. There is nothing here that implicates BIOS or OS. There is no reason to believe that write request would be from an OS or a BIOS. Why would a BIOS be making write requests? The questions seem totally irrelevant to anything pertinent to the application. It does not matter what made the write request, any write requests are handled in the way described.

Then the question is asked "Where if an OS driver requests?" Where makes no sense. Then the question is asked "If the logging refers to the OS cache write entries or if the BIOS request such a log accomplishes for as by definition BIOS writes are immediate and not cached within an OS cache." It does not matter what the write requests are accomplishing. The application is explaining how the write requests are handled. The write request could be accomplishing anything. If it is a BIOS request, what such log is accomplished is clearly explained in the specification. Further, the specification expands at page 6, line 4, that after being logged, the write requests are executed. Thus, one purpose of the log is obviously to list write requests to be executed.

The assertion that BIOS writes are immediate and not cached within the OS cache, if true, would suggest that what is being cached here are not BIOS write requests. Thus, the Examiner's question answers his own previous question to the extent the assertions made in the objections are correct in the first case. Further, it is said that nowhere is it remotely clear what the purpose is of the disclosed call stack manipulation as the BIOS call purpose and semantics are unambiguous and seemingly not relevant for OS driver calls. Of course, the confusion about OS versus BIOS is nowhere driven by anything in the specification, but is simply pulled out of thin air. As to the call stack manipulation, it is not clear what this is referring to. There is no call stack referred to within the application.

If the objection is to be maintained, an interview with the Examiner and his supervisor is hereby expressly requested. Please call the undersigned to set up such an interview.

With respect to the prior art rejection, rejection is plainly improper on its face. The alternative 102 or 103 rejection is legally improper. If the rejection is a 102 rejection, then it should so be stated. If there is something that is missing and the rejection seems to concede as much, then a 103 rejection should be stated. But a single reference Section 103 rejection is plainly improper since the reference by itself cannot teach any reason to modify itself.

Therefore, under either grounds, by admission and on its face, the rejection is improper. Further, the rejection is extremely unclear. The first sentence of the rejection makes no sense. It suggests that "thereby rendering the two equivalent" is unclear because it is not clear what the Examiner is saying is equivalent or what in the ensuing language is common knowledge. The rejection is so unclear as to preclude any response. The assertion of common knowledge to those of ordinary skill in the art is unclear because it is not clear what is alleged to be common knowledge. The "correspondingly logical inherent and rendering obvious" makes no sense either. The rejection is further deficient because it makes no effort whatsoever to address the language in the claims. It simply talks about a bunch of things that are not claimed and then attempts to make assertions about them with no reference to anything that is relevant to the claimed invention. The citation to the application that is co-pending to Garney is noted, but there is no suggestion as to how this application constitutes prior art other than Section 103(a), which obviously could not possibly apply. Further, there is not a single citation to anything within Garney, but simply assertions as to what Garney shows without citation.

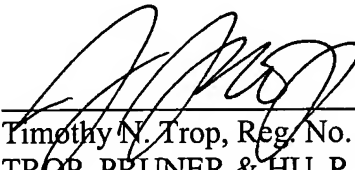
The assertion of Section 102(a) for the Applicant's own pending application seems impossible. Therefore, the rejection is respectfully traversed.

The assertion under Section 103 is barred by 103(c) since, at the time the invention of this application was made, both this application and the cited application were owned by Intel Corporation.

In view of these remarks, reconsideration is requested.

Respectfully submitted,

Date: April 25, 2006



---

Timothy N. Trop, Reg. No. 28,994  
TROP, PRUNER & HU, P.C.  
8554 Katy Freeway, Ste. 100  
Houston, TX 77024  
713/468-8880 [Phone]  
713/468-8883 [Fax]

Attorneys for Intel Corporation